



Euler-Bernoulli corotational beams in Chrono::Engine

Alessandro Tasora
alessandro.tasora@unipr.it

April 7, 2016

Abstract

This document presents some insights on the theory of corotational finite elements of Euler-Bernoulli type as implemented in **Chrono::Engine**. This type of finite element can be used for beams with large displacements.

1. Introduction

Among the various methods that allow finite element simulations with large geometric deformations, the corotational approach is one the most versatile because it fosters the reuse of formulas already developed for classical linear finite elements, that are widely available in literature.

We recall that in this context the corotational approach allows large displacements but it requires that strains must be small.

In this document we explain the theory behind the `chrono::fea::ChElementBeamEuler` finite element.

These finite elements are based on corotated Eulero-Bernoulli elements. In the following section we will briefly recall the basics of the Eulero-Bernoulli linear beam theory in 3D, and we will explain the corotational approach to use them in large displacements.

2. Eulero-Bernoulli three-dimensional beams

The theory of Eulero-Bernoulli beams is explained in many textbooks, here we recall only the main formulas. We make the assumption that material points on the normal to the mid-line remain on the normal during the deformation.

We define the properties of the section of the beam with the following parameters: area moments of inertia I_{zz} and I_{yy} , Young modulus E , shear modulus

G , area A , torsion constant J .

Note that shear is not considered in Euler-Bernoulli theory: these beams are well suited for problems with thin slender beams, and are less fit for thick short beams where the shear effects could be relevant. This said, we still introduced the shear modulus G , but this is required only for the torsional effects.

In the plane case, the following constitutive equations are used ¹, with (assumed small) deflection w along its abscissa, transverse load q and abscissa ς :

$$\frac{d^2}{d\varsigma^2} \left(EI \frac{d^2 w}{d\varsigma^2} \right) = q$$

This holds in the two transverse directions y and z , then similar constitutive equations exist for torsion load on x and axial load on x . The state of a finite element of a typical Euler-Bernoulli beam, as seen in a the element local coordinate system F , depends on the following vector that contains the local displacements and local rotations of the two A and B end nodes:

$$\underline{\mathbf{d}} = \begin{pmatrix} \underline{\mathbf{d}}_A \\ \underline{\boldsymbol{\theta}}_A \\ \underline{\mathbf{d}}_B \\ \underline{\boldsymbol{\theta}}_B \end{pmatrix} \quad (1)$$

This is used to get the local deflection w along the abscissa (the local x direction), once $\underline{\mathbf{d}}$ is given, by using the typical cubic Hermite shape functions $N(\varsigma)$ (that we do not report for sake of space) as in:

$$w_y(\varsigma) = \begin{pmatrix} N_{d_A}(\varsigma) & N_{\theta_A}(\varsigma) & N_{d_B}(\varsigma) & N_{\theta_B}(\varsigma) \end{pmatrix} \begin{pmatrix} \underline{\mathbf{d}}_{A,y} \\ \underline{\boldsymbol{\theta}}_{A,y} \\ \underline{\mathbf{d}}_{B,y} \\ \underline{\boldsymbol{\theta}}_{B,y} \end{pmatrix} \quad (2)$$

Same for z deflection. For torsion and axial deformation, shape functions are just linear interpolations and we do not report them.

Putting all together, for both y and z transverse directions, and considering axial and torsional effects, with a matrix of shape functions S , local deflection $\underline{\mathbf{d}}(\varsigma)$ and local slope $\underline{\boldsymbol{\theta}}(\varsigma)$ are:

$$\begin{pmatrix} \underline{\mathbf{d}}(\varsigma) \\ \underline{\boldsymbol{\theta}}(\varsigma) \end{pmatrix} = [S] \underline{\mathbf{d}} \quad (3)$$

Derivatives of shape functions can also be used to express local curvature $\frac{d^2 w}{d\varsigma^2}$. This can be useful also because, for post-processing, once the curvature $\frac{d^2 w}{d\varsigma^2}$ is known, the internal torque M at a section is easily computed as

$$M = -EI \frac{d^2 w}{d\varsigma^2}$$

¹here expressed without the inertial terms for simpler expression

For reference, we report here the local stiffness matrix for a three dimensional Eulero-Bernoulli beam, whose degrees of freedom are arranged as in the \underline{d} vector:

$$\underline{\mathbf{K}} = \begin{pmatrix} k_u & 0 & 0 & 0 & 0 & 0 & -k_u & 0 & 0 & 0 & 0 & 0 \\ & k_v & 0 & 0 & 0 & k_{v\psi} & 0 & -k_v & 0 & 0 & 0 & k_{v\psi} \\ & & k_w & 0 & -k_{w\theta} & 0 & 0 & 0 & -k_w & 0 & -k_{w\theta} & 0 \\ & & & k_\phi & 0 & 0 & 0 & 0 & 0 & -k_\phi & 0 & 0 \\ & & & & k_\theta & 0 & 0 & 0 & k_{w\theta} & 0 & k_{\theta\theta} & 0 \\ & & & & & k_\psi & 0 & -k_{v\psi} & 0 & 0 & 0 & k_{\psi\psi} \\ & & & & & & k_u & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & k_v & 0 & 0 & 0 & -k_{v\psi} \\ \text{Sym.} & & & & & & & & k_w & 0 & k_{w\theta} & 0 \\ & & & & & & & & & k_\phi & 0 & 0 \\ & & & & & & & & & & k_\theta & 0 \\ & & & & & & & & & & & k_\psi \end{pmatrix} \quad (4)$$

where $k_w = \frac{12EI_{yy}}{L^3}$, $k_\theta = \frac{4EI_{yy}}{L}$, $k_{w\theta} = \frac{6EI_{yy}}{L^2}$, $k_{\theta\theta} = \frac{2EI_{yy}}{L}$, $k_v = \frac{12EI_{zz}}{L^3}$, $k_\psi = \frac{4EI_{zz}}{L}$, $k_{v\psi} = \frac{6EI_{zz}}{L^2}$, $k_{\psi\psi} = \frac{2EI_{zz}}{L}$, $k_u = \frac{EA}{L}$, $k_\phi = \frac{GJ}{L}$.
 $k_w = 12EI_{yy}/L^3$, $k_\theta = 4EI_{yy}/L$, $k_{w\theta} = 6EI_{yy}/L^2$, $k_{\theta\theta} = 2EI_{yy}/L$, $k_v = 12EI_{zz}/L^3$, $k_\psi = 4EI_{zz}/L$, $k_{v\psi} = 6EI_{zz}/L^2$, $k_{\psi\psi} = 2EI_{zz}/L$, $k_u = EA/L$, $k_\phi = GJ/L$.

3. Advanced beam sections

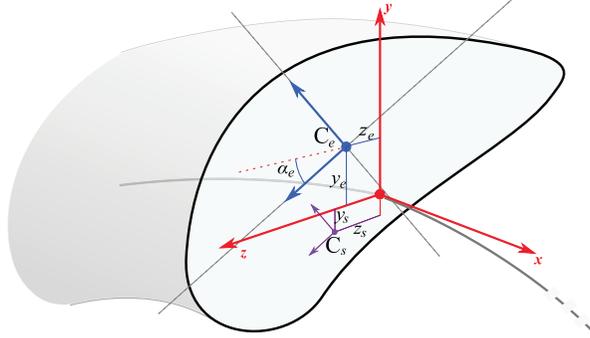


Figure 1: Elastic center C_e for a generic section with arbitrary shape; moments of inertia I_{yy} and I_{zz} could be expressed relative to axes being rotated by an angle α_e . The shear center C_s often coincides with C_e in simple symmetric sections.

The expression of Eq. (4) is valid for symmetric sections whose shear center C_s and elastic center C_e are coincident with the centerline reference. A more general case is depicted in Fig. 1. Denoting with y_s and z_s the offsets of the shear

center, with y_e and z_e the offsets of the elastic center, with α_e the rotation of the axes used for obtaining of I_{yy} and I_{zz} , and assuming that those parameters are constant through the length of the beam, one can transform the local matrix $\underline{\mathbf{K}}$ as:

$$\underline{\mathbf{K}}' = \mathbf{T}_s \mathbf{T}_c \mathbf{T}_r \underline{\mathbf{K}} \mathbf{T}_r^t \mathbf{T}_c^t \mathbf{T}_s^t \quad (5)$$

where

$$\mathbf{T}_r = \begin{pmatrix} \mathbf{R}_\alpha & & & \\ & \mathbf{R}_\alpha & & \\ & & \mathbf{R}_\alpha & \\ & & & \mathbf{R}_\alpha \end{pmatrix} \mathbf{T}_r = \text{blockdiag}(\mathbf{R}_\alpha \ \mathbf{R}_\alpha \ \mathbf{R}_\alpha \ \mathbf{R}_\alpha)$$

with

$$; \quad \mathbf{R}_\alpha = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_e) & -\sin(\alpha_e) \\ 0 & \sin(\alpha_e) & \cos(\alpha_e) \end{pmatrix} \quad \exp(\alpha_e \mathbf{e}_1 \times)$$

and

$$\mathbf{T}_c = \begin{pmatrix} \mathbf{T}_{c_A} & \mathbf{0}_{6 \times 6} \\ \mathbf{0}_{6 \times 6} & \mathbf{T}_{c_B} \end{pmatrix}; \quad \mathbf{T}_{c_A} = \mathbf{T}_{c_B} = \begin{pmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{T}_{c_{zy}} & \mathbf{I}_{3 \times 3} \end{pmatrix}; \quad \mathbf{T}_{c_{zy}} = \begin{pmatrix} 0 & 0 & 0 \\ c_z & 0 & 0 \\ -c_y & 0 & 0 \end{pmatrix} \quad (6)$$

and

$$\mathbf{T}_s = \begin{pmatrix} \mathbf{T}_{s_A} & \mathbf{0}_{6 \times 6} \\ \mathbf{0}_{6 \times 6} & \mathbf{T}_{s_B} \end{pmatrix}; \quad \mathbf{T}_{s_A} = \mathbf{T}_{s_B} = \begin{pmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{T}_{s_{zy}} & \mathbf{I}_{3 \times 3} \end{pmatrix}; \quad \mathbf{T}_{s_{zy}} = \begin{pmatrix} 0 & s_z & -s_y \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (7)$$

Note that the stiffness matrix of Eq. (4), or Eq. (5), is expressed here in symbolic form only for the simple case of constant section and constant material; for more generic cases (ex. varying section) one should compute $\underline{\mathbf{K}}$ by Gauss quadrature.

4. Corotated beams

Figure 2 shows the concept of the corotational formulation in Chrono::Engine. A floating coordinate system F follows the deformed element, thus the overall gross motion into the deformed state \mathcal{C}_D can be seen as the superposition of a large rigid body motion from the reference configuration \mathcal{C}_0 to the so called *floating* or *shadow* configuration \mathcal{C}_S , plus a local small-strain deformation from \mathcal{C}_S to \mathcal{C}_D . In this work, underlined symbols will represent variables expressed in the basis of the floating reference F .

The rationale of the corotational approach is a procedure to compute a global tangent stiffness $\underline{\mathbf{K}}_e$ and a global force $\underline{\mathbf{f}}_e$ for each element e , given its local $\underline{\mathbf{K}}$, its local $\underline{\mathbf{f}}$ and the rigid body motion of the frame F in \mathcal{C}_0 to F in \mathcal{C}_S .

Whenever the element moves, the position and rotation of F is updated. In literature there are many options to this end; to avoid dependence on connectivity [7], in our implementation we decided to put the origin of F in the

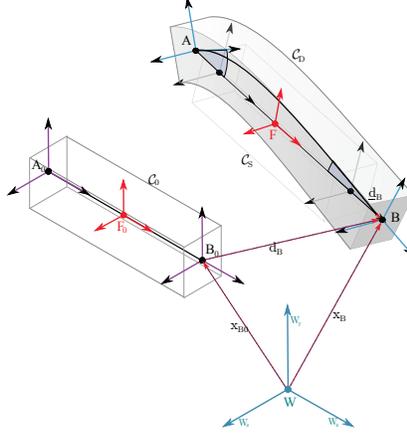


Figure 2: A schematic representation of the corotational concept.

midpoint of the AB segment, as $\mathbf{x}_F = \frac{1}{2}(\mathbf{x}_B - \mathbf{x}_A)$, and we align its \mathbf{X} axis with $\mathbf{x}_B - \mathbf{x}_A$. The remaining \mathbf{Y} and \mathbf{Z} axes of F are obtained with a Gram-Schmidt orthogonalization, enforcing \mathbf{Y} to bisect the \mathbf{Y} axes of A and B when projected on the plane orthogonal to \mathbf{X} . This is important in case of torsion.

The rotation matrix of F is $\mathbf{R}_F \in \text{SO3}$; it is parametrized with the unit quaternion $\rho_F \in \mathbb{H}_1$. Similarly, quaternions are used to store the rotations of the two nodes, with ρ_A and ρ_B . Quaternions are used also to define the rotation of rigid bodies in the system, if any. For this reason it is also easy to 'connect' an Euler beam end to a body, for example using the `chrono::ChLinkMate` constraints.

The state of a system with beams is $\mathbf{s} = [\mathbf{q}, \mathbf{v}]$ with $\mathbf{q} = [\mathbf{x}_1, \rho_1, \mathbf{x}_2, \rho_2, \dots, \mathbf{x}_n, \rho_n] \in \mathbb{R}^{(3+4)n}$ and $\mathbf{v} = [\mathbf{v}_1, \omega_1, \mathbf{v}_2, \omega_2, \dots, \mathbf{v}_n, \omega_n] \in \mathbb{R}^{(3+3)n}$. Note that, because of some algorithmic optimizations, we consider ω_i to be expressed in the *local* basis of the i -th node unlike $\mathbf{x}_i, \rho_i, \mathbf{v}_i$ that are considered in the global basis: this is different from implementations in many other simulation codes.

For each element, given the global positions and rotations of the two end nodes, stored in the \mathbf{s} state at indexes i_A and i_B , it is possible to compute the actual displacement part of $\underline{\mathbf{d}}$ as $\underline{\mathbf{d}}_A = \underline{\mathbf{x}}_A - \underline{\mathbf{x}}_{A_0} = \mathbf{R}_{F_0}^t(\mathbf{x}_{A_0} - \mathbf{x}_{F_0}) - \mathbf{R}_F^t(\mathbf{x}_A - \mathbf{x}_F)$ and $\underline{\mathbf{d}}_B = \underline{\mathbf{x}}_B - \underline{\mathbf{x}}_{B_0} = \mathbf{R}_{F_0}^t(\mathbf{x}_{B_0} - \mathbf{x}_{F_0}) - \mathbf{R}_F^t(\mathbf{x}_B - \mathbf{x}_F)$.

The rotation part, however, introduces a complication, owing to the fact that finite rotations do not compose as vectors and must be dealt with special algebraic tools. First, one must compute the local rotation of nodes respect to the F floating reference with $\underline{\mathbf{R}}_A = \mathbf{R}_F^t \mathbf{R}_A \mathbf{R}_{A_\Delta}^t$ and $\underline{\mathbf{R}}_B = \mathbf{R}_F^t \mathbf{R}_B \mathbf{R}_{B_\Delta}^t$. Equivalently, one can use quaternion products to write: $\underline{\rho}_A = \rho_F^t \rho_A \rho_{A_\Delta}^t$ and $\underline{\rho}_B = \rho_F^t \rho_B \rho_{B_\Delta}^t$. Here the optional term $\rho_{A_\Delta}^t$ (or $\mathbf{R}_{A_\Delta}^t$) represents the initial rotation of node A respect to F_0 .

Then, the finite rotation pseudovectors $\underline{\boldsymbol{\theta}}_A$ and $\underline{\boldsymbol{\theta}}_B$ are obtained in the follow-

ing way. It is known that, for an element \mathbf{R} in Lie group $\text{SO}(3)$ and an element Θ in the corresponding Lie algebra $\text{so}(3)$, one has $\Theta = \text{skew}(\theta)$ where θ is also an element of the Lie group $\text{Spin}(3)$, double cover of $\text{SO}(3)$. Vice versa, one can extract the θ vector from the Θ spinor by computing $\theta = \text{axial}(\Theta)$. Also, it holds $\mathbf{R} = \exp(\Theta)$ and $\Theta = \text{Log}(\mathbf{R})$, where $\exp(\cdot)$ builds the rotation matrix using an exponential; for details on this exponential and the implementation of $\text{skew}(\cdot)$, $\text{axial}(\cdot)$, $\text{Log}(\cdot)$, see for example [11].

In the work of other Authors, the theory above is used to compute $\underline{\theta}_A = \text{axial}(\text{Log}(\underline{\mathbf{R}}_A))$, but in our case the adoption of quaternions lead to an alternative, more straightforward expression. In fact it is known that for a unit quaternion $\rho \in \mathbb{H}_1$ it holds $\rho = [\cos(\theta/2), \mathbf{u} \sin(\theta/2)]$, with rotation angle $\theta = |\theta|$ about rotation unit vector $\mathbf{u} = \theta/\theta$. Therefore it is possible to compute $\underline{\theta}_A$ and $\underline{\theta}_B$ as: $\theta_A = 2 \arccos(\Re(\rho_A))$, $\mathbf{u}_A = \frac{1}{\sin(\theta_A/2)} \Im(\rho_A)$, and $\underline{\theta}_A = \theta_A \mathbf{u}_A$ (the same for the B node).

Once $\underline{\mathbf{d}} = [\underline{\mathbf{d}}_A, \underline{\theta}_A, \underline{\mathbf{d}}_B, \underline{\theta}_B]$ has been computed, well-known theories are available to compute the stiffness matrix $\underline{\mathbf{K}} = \underline{\mathbf{K}}(\underline{\mathbf{d}})$. In this work we compute $\underline{\mathbf{K}}$ using the Eulero-Bernoulli theory, and in general we set $\underline{\mathbf{f}}_{in} = \underline{\mathbf{K}}\underline{\mathbf{d}}$.

The local data $\underline{\mathbf{K}}$ and $\underline{\mathbf{f}}_{in}$ must be mapped to the global reference: to this end we use the corotational approach expressed in [5], where the adoption of projectors that filter rigid body motion is used to improve the consistency and the convergence of the method. Such formulation requires the introduction of various matrices, in the following we succinctly report them, along with modifications that we use in our method.

- the $\Lambda(\theta) = \frac{\partial \theta}{\partial \omega}$ matrix, whose analytic expression is $\Lambda(\theta) = \mathbf{I}_{3 \times 3} - \frac{1}{2} \text{skew}(\theta) + \zeta \text{skew}(\theta)^2$ with $\zeta = (1 - \frac{1}{2} \theta \cotan(\frac{1}{2} \theta)) / \theta^2$,
- the \mathbf{H} transformation matrix:

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_n(\underline{\theta}_A) & \mathbf{0}_{6 \times 6} \\ \mathbf{I}_{6 \times 6} & \mathbf{H}_n(\underline{\theta}_B) \end{pmatrix}; \quad \mathbf{H}_n(\theta) = \begin{pmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \Lambda(\theta) \end{pmatrix} \quad (8)$$

that tends to a unit matrix $\mathbf{I}_{12 \times 12}$ for $\theta \downarrow 0$,

- the $\underline{\mathbf{P}}$ projector matrix: $\underline{\mathbf{P}} = \mathbf{I}_{12 \times 12} - \underline{\mathbf{S}}^D \underline{\mathbf{G}}$ where $\underline{\mathbf{S}}^D$ is the so called *spin lever* matrix, built with \mathbf{x}_A and \mathbf{x}_B , the positions of the end nodes respect to the center F of the beam, expressed in F basis:

$$\underline{\mathbf{S}}^D = \begin{pmatrix} -\text{skew}(\mathbf{x}_A) \\ \mathbf{I}_{3 \times 3} \\ -\text{skew}(\mathbf{x}_B) \\ \mathbf{I}_{3 \times 3} \end{pmatrix} \quad (9)$$

and where $\underline{\mathbf{G}}$ is the so called *spin fitter* matrix, that takes into account the change of orientation of the F frame as the end nodes change position or rotation. For the two nodes beam, it is $\underline{\mathbf{G}} = [\partial \omega_F / \partial \mathbf{x}_A, \partial \omega_F / \partial \omega_A, \dots]$, and for our custom choice of orientation and position of F , described at

the beginning of this section, we have

$$\underline{\mathbf{G}} = \left[\begin{array}{ccc|ccc|ccc|ccc} 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/L & 0 & 0 & 0 & 0 & 0 & -1/L & 0 & 0 & 0 \\ 0 & -1/L & 0 & 0 & 0 & 0 & 0 & 1/L & 0 & 0 & 0 & 0 \end{array} \right] \quad (10)$$

Note that this expression is different from the one reported in [10] because they put the F frame at the beginning of the beam whereas we put it in the middle, moreover it rotates a bit differently,

- the \mathbf{R}_\diamond rotation-transformation matrix:

$$\mathbf{R}_\diamond = \begin{pmatrix} \mathbf{R}_F & & & \\ & \mathbf{R}_A^t \mathbf{R}_F & & \\ & & \mathbf{R}_F & \\ & & & \mathbf{R}_B^t \mathbf{R}_F \end{pmatrix} \quad (11)$$

Note that \mathbf{R}_\diamond is different from the one often reported in literature, ex. in in [10] or [5], because we update the rotation of nodes with rotation pseudovectors expressed in node *local* coordinates (hence the \mathbf{R}_A^t and \mathbf{R}_B^t transformation), coherently with what we said about angular velocities being expressed in local references and not in global reference, in our state \mathbf{s} .

Given the matrices above, one can compute the global version of internal forces:

$$\underline{\mathbf{f}}_{in} = \mathbf{R}_\diamond \underline{\mathbf{P}}^t \mathbf{H}^t \underline{\mathbf{f}}_{in} \quad (12)$$

For the computation of the global tangent stiffness, one needs two additional matrices. First, we split the last part of Eq. (12) in four three-dimensional vectors: $\underline{\mathbf{P}}^t \mathbf{H}^t \underline{\mathbf{f}}_{in} = [\underline{\mathbf{n}}_A, \underline{\mathbf{m}}_A, \underline{\mathbf{n}}_B, \underline{\mathbf{m}}_B]$, then we build the $\underline{\mathbf{F}}_{nm}$ and $\underline{\mathbf{F}}_n$ matrices:

$$\underline{\mathbf{F}}_{nm} = \begin{pmatrix} \text{skew}(\underline{\mathbf{n}}_A) \\ \text{skew}(\underline{\mathbf{m}}_A) \\ \text{skew}(\underline{\mathbf{n}}_B) \\ \text{skew}(\underline{\mathbf{m}}_B) \end{pmatrix}, \quad \underline{\mathbf{F}}_n = \begin{pmatrix} \text{skew}(\underline{\mathbf{n}}_A) \\ \mathbf{0}_{3 \times 3} \\ \text{skew}(\underline{\mathbf{n}}_B) \\ \mathbf{0}_{3 \times 3} \end{pmatrix} \quad (13)$$

Finally one can compute the tangent stiffness matrix of the element in global coordinates, also accounting for geometric stiffening:

$$\mathbf{K} = \mathbf{R}_\diamond (\underline{\mathbf{P}}^t \mathbf{H}^t \underline{\mathbf{K}} \mathbf{H} \underline{\mathbf{P}} - \underline{\mathbf{F}}_{nm} \underline{\mathbf{G}} - \underline{\mathbf{G}}^t \underline{\mathbf{F}}_n \underline{\mathbf{P}} + \underline{\mathbf{P}}^t \underline{\mathbf{L}}_H \underline{\mathbf{P}}) \mathbf{R}_\diamond^t \quad (14)$$

$$\mathbf{K} = \mathbf{R}_\diamond (\mathbf{K}_M - \mathbf{K}_{GR} - \mathbf{K}_{GP} + \mathbf{K}_{GH}) \mathbf{R}_\diamond^t \quad (15)$$

We remark the following notes:

- the three terms \mathbf{K}_{GR} (related to change in rotation of the F frame), \mathbf{K}_{GP} (related to changes in projectors), \mathbf{K}_{GH} (related to changes in \mathbf{H}) are responsible of the so called *geometric stiffness*,

- the \mathbf{K}_{GH} term is not used in our formulation since we found no major benefits in computing it; see [5] for details on \mathbf{L}_H ,
- the \mathbf{K}_M , which represents the so called *material stiffness*, is always symmetric (at least with Eulero-Bernoulli beams), but the terms for geometric stiffness introduce asymmetry; this can be a major drawback because many iterative solvers benefit from symmetric matrices ²;
- some Authors [9] show that, under mild assumptions, neglecting the asymmetric part does not hamper the convergence of Newton-Raphson iterations; hence the variant: $\mathbf{K}_{symm} = \mathbf{R}_\diamond (\mathbf{P}^t \mathbf{H}^t \mathbf{K} \mathbf{H} \mathbf{P} - \mathbf{F}_{sy} \mathbf{G} - \mathbf{G}^t \mathbf{F}_{sy}^t \mathbf{P}) \mathbf{R}_\diamond^t$ where $\mathbf{F}_{sy} = \frac{1}{2}(\mathbf{F}_{nm} + \mathbf{F}_n)$

5. Benchmark: Princeton beam experiment

This benchmark aims at the validation of the finite element implementations in a static problem with geometric nonlinearity. A thin cantilevered beam, constrained in O , is subject to large deformations and large rotations because of a tip load in E ; for different angles θ one obtains out-of-plane displacements even if the load is vertical, and the beam is subject to a twisting action.

Experimental results, for a beam made with 7075 aluminium, are available in [3, 4] and are used for comparison.

We list the main properties, with reference to Figure 3: beam length $L = 0.508\text{m}$, section thickness $T = 3.2024\text{mm}$, section height $H = 12.77\text{mm}$, Young modulus $E = 71.7\text{GPa}$, $\nu = 0.31$, $G = E \frac{(1+\nu)}{2} = 27.37\text{GPa}$.

Three loading conditions are tested: $P_1 = 4.448\text{N}$, $P_2 = 8.896\text{N}$, and $P_3 = 13.345\text{N}$, for increasing values of the θ angle in the $[0^\circ, 90^\circ]$ range.

Results in Figs. 4, 5 and 6 show a good agreement between the present corotational beam formulation and the geometrically-exact beam formulations presented in [2] for Dymore and in [6] for MBDyn, as well as an agreement with the experimental results in [3, 4], which was recently discussed in [1].

We remark that, because of the geometric nonlinearity, the solver has to perform few Newton-Raphson steps before obtaining a zero residual. For very large nonlinearities, a continuation strategy might help the convergence of the Newton-Raphson solver.

6. Benchmark: Lateral buckling

This benchmark tests nonlinear effects in a dynamic context. A beam is bent in its plane of greatest flexural rigidity, up to the point that triggers lateral

²For instance, the MINRES and CG Krylov solvers expect Hermitian matrices, whereas unsymmetric matrices would require more advanced solvers like GCR or GMRES. The latter cannot exploit the three-term recurrency feature as in Hermitian matrices, hence restarting is needed to save memory. Incidentally, we experienced that, in some problems, restarting alone is not sufficient because GMRES or GCR can be trapped in cycling patterns, so more complex restarting strategies with deflating and anti-stagnation must be adopted.

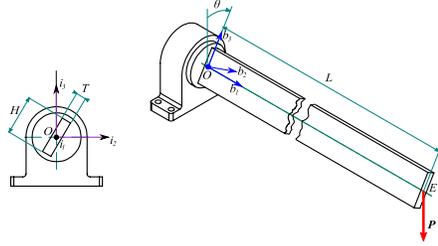


Figure 3: Setup of the benchmark for the Princeton beam experiment.

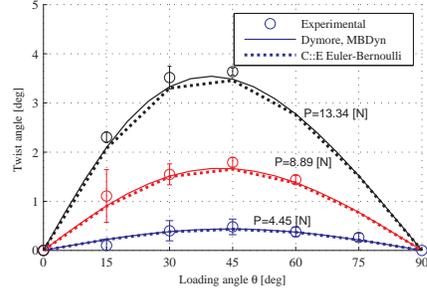


Figure 4: Twist rotation of the beam for the Princeton experiment.

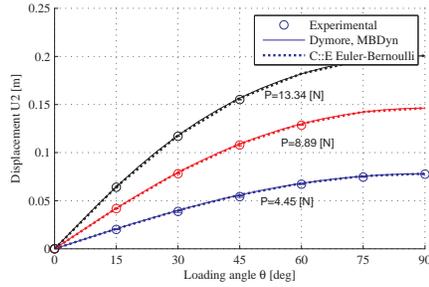


Figure 5: Flapwise displacement at the beam tip versus loading angle for three loading conditions.

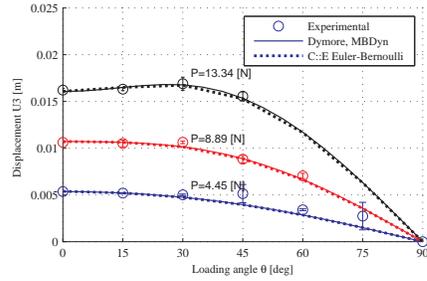


Figure 6: Chordwise displacement at the beam tip versus loading angle for three loading conditions.

buckling. In a quasi-static non-linear analysis, results are visible in Fig. 8. In the context of dynamics, when buckling occurs, the beam snaps laterally and twists, inducing highly oscillatory motions. The corotational approach can capture the nonlinear nature of this phenomena.

As shown in Fig. 7, the RC beam is clamped at point R , its length is $L = 1\text{m}$, and its rectangular section has size $H = 100\text{mm}$ and $B = 10\text{mm}$.

To induce the snapping, a tip load at C is imposed by mean of a rotating crank GB and a vertical rod TB , with a spherical joint in C and a revolute joint in B . An initial imperfection is simulated by displacing the vertical bar and the crank by an offset $d = 0.1\text{mm}$ in the off-plane direction i_2 . The crank has length $L_c = 0.05\text{m}$ and a circular section with diameter $D_r = 24\text{mm}$, while the vertical rod has a length $L_r = 0.25\text{m}$ and a circular section with diameter $D_r = 48\text{mm}$. The rotation of the crank is initially enforced by a prescribed motion function $\phi_c(t) = \pi(1 - \cos(\pi t/T_c))/2$, with $T_c = 0.4\text{s}$, then for $t > T_c$ it is $\phi_c(t) = \pi$.

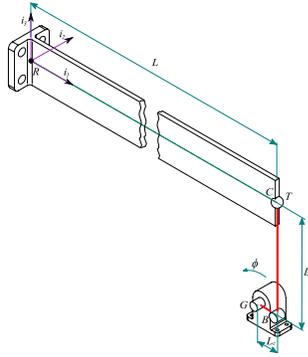


Figure 7: Setup of the benchmark for lateral buckling dynamics.

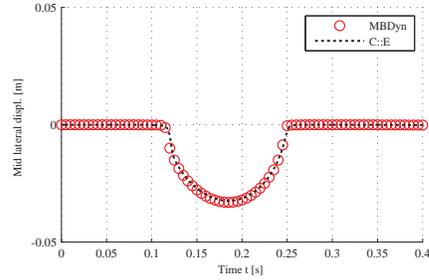


Figure 8: Static displacement of the beam along i_2 , at the mid point.

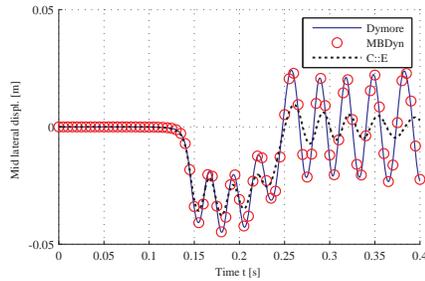


Figure 9: Displacement of the beam along i_2 , at the mid point.

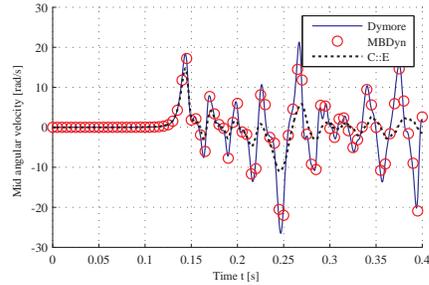


Figure 10: Angular velocity of the beam, at the mid point.

All parts are made of aluminum, hence with Young modulus $E = 73\text{GPa}$ and Poisson ratio $\nu = 0.3$. Given the above mentioned sections, their inertia values I_{zz} and I_{yy} and their torsion constants J are computed using formulas available in classical textbooks.

In the Chrono::Engine test, the RC beam is modeled with 12 finite elements, whereas the crank and the rod are modeled with 3 elements each. Results in Figs. 9 and 10 show that the lateral buckling is triggered exactly at the same moment for all the formulations, although the Chrono::Engine integral is more damped. The numerical damping is a consequence of the fact that the Chrono::Engine default integrator is a timestepper for DVI non-smooth problems [8, 12]. This, in the case of no frictional contacts, boils down to a linearly-implicit first-order scheme, hence it shows the same damping effect of an implicit Euler method. Other integrals are obtained with higher order methods and are affected by numerical damping to a much lower degree.

7. Conclusion

Quaternions are extensively used in **Chrono::Engine** to work with rotations. Look into the documentation of the the `chrono::ChQuaternion` class for additional information.

More information available in **Chrono::Engine** web site.

References

- [1] O. A. Bauchau, G. Wu, P. Betsch, A. Cardona, J. Gerstmayr, B. Jonker, P. Masarati, and V. Sonneville. Validation of flexible multibody dynamics beam formulations using benchmark problems. In *IMSD 2014*, Korea, June 30-July 3 2014.
- [2] Olivier A. Bauchau and N. K. Kang. A multibody formulation for helicopter structural dynamic analysis. *Journal of the American Helicopter Society*, 38(2):3–14, 1993.
- [3] E. H. Dowell and J. J. Traybar. An experimental study of the nonlinear stiffness of a rotor blade undergoing flap, lag, and twist deformations. Aerospace and Mechanical Science Report 1194, Princeton University, January 1975.
- [4] E. H. Dowell and J. J. Traybar. An experimental study of the nonlinear stiffness of a rotor blade undergoing flap, lag, and twist deformations. Aerospace and Mechanical Science Report 1257, Princeton University, December 1975.
- [5] C.A. Felippa and B. Haugen. A unified formulation of small-strain corotational finite elements: I. theory. *Computer Methods in Applied Mechanics and Engineering*, 194(2124):2285 – 2335, 2005. Computational Methods for Shells.
- [6] Gian Luca Ghiringhelli, Pierangelo Masarati, and Paolo Mantegazza. A multi-body implementation of finite volume beams. *AIAA Journal*, 38(1):131–138, January 2000.
- [7] P. Masarati and M. Morandini. Intrinsic deformable joints. *Multibody System Dynamics*, 23(4):361–386, 2010. doi:10.1007/s11044-010-9194-y.
- [8] Dan Negrut, Alessandro Tasora, Hammad Mazhar, Toby Heyn, and Philipp Hahn. Leveraging parallel computing in multibody dynamics. *Multibody System Dynamics*, 27:95–117, 2012.
- [9] B. Nour-Omid and C.C. Rankin. Finite rotation analysis and consistent linearization using projectors. *Computer Methods in Applied Mechanics and Engineering*, 93(3):353 – 384, 1991.

- [10] C.C. Rankin and B. Nour-Omid. The use of projectors to improve finite element performance. *Computers & Structures*, 30(12):257 – 267, 1988.
- [11] J.C. Simo and L. Vu-Quoc. A three-dimensional finite-strain rod model. part ii: Computational aspects. *Computer Methods in Applied Mechanics and Engineering*, 58(1):79 – 116, 1986.
- [12] Alessandro Tasora and Mihai Anitescu. A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):439 – 453, 2011.